



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants: Doolittle et al.

Group Art Unit: 2141

Serial No.: 09/443,202

Examiner: Stephan F. Willett

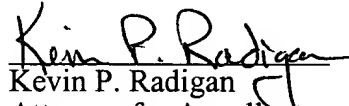
Filed: 11/18/99

Appeal No.:

For: METHOD, SYSTEM AND PROGRAM PRODUCTS FOR MANAGING THREAD  
POOLS OF A COMPUTING ENVIRONMENT TO AVOID DEADLOCK  
SITUATIONS BY DYNAMICALLY ALTERING ELIGIBLE  
THREAD POOLS

Certificate of Mailing

I hereby certify that this correspondence is being deposited with  
the U.S. Postal Service as first class mail in an envelope addressed  
to: Mail Stop Appeal Brief – Patents, Commissioner for Patents,  
P.O. Box 1450, Alexandria, VA 22313-1450, on August 31,  
2004.

  
Kevin P. Radigan  
Attorney for Appellants  
Registration No. 31,789

Date of Signature: August 31, 2004

Mail Stop Appeal Brief – Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

09/02/2004 JBALINAN 00000046 090457 09443202

01 FC:1402 330.00 DA

Brief of Appellants

Dear Sir:

This is an appeal from a final rejection, dated April 5, 2004, rejecting claims 1, 3, 4, 9-11, 14-18, 21-22, 27, 29, 30, 35-37, 40-44, 47-48, 53-56, 58, 60, 61, 66-68, 70-75, 78 & 79, all the claims being considered in the above-identified application. This Brief is accompanied by a transmittal letter authorizing the charging of Appellants' deposit account for payment of the requisite fee set forth in 37 C.F.R. §1.17(c).

EN999058

### **Real Party In Interest**

This application is assigned to **International Business Machines Corporation** by virtue of an assignment executed by the inventors on November 16, 1999 and November 17, 1999, and recorded with the United States Patent and Trademark Office at reel 010407, frame 0357, on November 18, 1999. Therefore, the real party in interest is **International Business Machines Corporation**.

### **Related Appeals and Interferences**

To the knowledge of the Appellants, Appellants' undersigned legal representative, and the assignee, there are no other appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the instant appeal.

### **Status of Claims**

This patent application was filed on November 18, 1999, with the U.S. Patent and Trademark Office. As filed, the application included eighty-three (83) claims, of which eight (8) were independent claims (i.e., claims 1, 23, 27, 49, 53, 57, 58 & 80).

In an initial Office Action dated June 5, 2002, a new title was required and claims 1-83 were subject to restriction under 35 U.S.C. §121 between Group I (claims 1-4, 9-11, 14-22, 27-30, 35-37, 40-48, 53-56, 58-61, 66-68 and 71-79) and Group II (claims 5-8, 12, 13, 23-26, 31-34, 38, 39, 49-52, 57, 62-65). Claims 1-4, 9-11, 14-22, 27-30, 35-37, 40-48, 53-56, 58-61, 66-68 and 71-79 of Group I were provisionally elected with traverse by Appellants' representative, Blanche Schiller, on May 15, 2002. The provisionally elected claims of Group I were rejected under 35 U.S.C. §103(a) as being unpatentable over Brobst et al (U.S. Patent No. 6,125,382; hereinafter "Brobst") in view of Schoening et al. (U.S. Patent No. 6,205,465; hereinafter "Schoening").

In response, Appellants' filed a Continued Prosecution Application dated September 5, 2002, in which a new title was provided, election of claims 1-4, 9-11, 14-22, 27-30, 35-37, 40-48, 53-56, 58-61, 66-68 and 71-79 was confirmed, and the restriction requirement was traversed. No claims were amended.

In an Office Action dated October 18, 2002, the restriction requirement was deemed proper; and claims 1-4, 9-11, 14-22, 27-30, 35-37, 40-48, 53-56, 58-61, 66-68 and 71-79 were rejected under 35 U.S.C. §103(a) as unpatentable over Schoening in view of Furlani et al. (U.S. Patent No. 5,995,998; hereinafter "Furlani"). In Appellants' response mailed January 21, 2003, no claims were amended.

In a final Office Action dated February 21, 2003, claims 1-4, 9-11, 14-22, 27-30, 35-37, 40-48, 53-56, 58-61, 66-68 and 71-79 were rejected under 35 U.S.C. §103(a) as unpatentable over Schoening in view of Furlani. In Appellants' response transmitted by facsimile on April 22, 2003, claims 1, 3, 27, 29, 53 & 58 were amended and claims 2, 5-8, 12-13, 23-26, 28, 31-34, 38-39, 49-52, 57, 59, 62-65, 69, 80-83 were cancelled.

Appellants did not receive an advisory action (or other form of reply) from the Patent Office to their response dated April 22, 2003. Further inquiry revealed that the case file could not be located at the Patent Office and, thus, Appellants' response was re-transmitted by facsimile on May 20, 2003. The case file was officially declared lost by the Patent Office on June 20, 2003. As a result, Appellants filed a Request for Continued Examination on July 21, 2003 (with a petition for two month extension of time), which requested consideration of the amendments submitted in Appellants' response dated April 22, 2003.

In an Office Action dated November 20, 2003, claims 1, 3-4, 9-11, 14-22, 27, 29-30, 35-37, 40-48, 53-56, 58, 60-61, 66-68 and 70-79 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schoening in view of Belkin et al. (U.S. Patent No. 6,542,920; hereinafter "Belkin"). In Appellants' response mailed February 19, 2004, claims 1, 27, 53 & 58 were amended and claims 19, 20, 45, 46, 76 & 77 were cancelled (without prejudice).

In a final Office Action dated April 5, 2004, independent claims 1, 27, 53 and 58<sup>1</sup> were rejected under 35 U.S.C. §112, first paragraph, as containing subject matter not described in the specification, as well as under 35 U.S.C. §112, second paragraph, as indefinite for failing to particularly point out and distinctly claim the subject matter regarded as the invention. Additionally, claims 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schoening in view of Belkin. In Appellants' response mailed May 28, 2004, no claims were amended.

A Notice of Appeal to the Board of Patent Appeals and Interferences was mailed on July 2, 2004.

In an Advisory Action dated July 9, 2004, Appellants' response was considered, and the §112 rejections were partially withdrawn. The Advisory Action indicated that the response mailed May 28, 2004 did not place the application in better form for appeal and or in condition for allowance.

The status of the pending claims is therefore as follows:

Claims allowed – none;

Claims objected to – none;

Claims rejected – 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79; and

Claims canceled – claims 2, 5-8, 12-13, 19-20, 23-26, 28, 31-34, 38-39, 45-46, 49-52, 57, 59, 62-65, 69, 76-77, 80-83

Appellants are appealing the rejection of claims 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79.

---

<sup>1</sup> The Office Action mistakenly refers to Claim "59" which was previously cancelled by Appellants in their response faxed April 22, 2003. Appellants submit that reference in the Office Action to claim "59" was an inadvertent, typographical error and the Examiner actually meant to recite claim "58" (still pending).

### **Status of Amendments**

Appellants proffered no amendments responsive to the final Office Action dated April 5, 2004. The claims as set out in the Appendix include all prior entered claim amendments.

### **Summary of the Invention**

Appellants' invention comprises a technique (e.g., 1, 27, 53 and 58) for managing thread pools of a computing environment (see FIGs. 1 & 2; page 9, lines 19<sup>+</sup>). This technique includes receiving from a first requester of the computing environment a request to be processed (see FIGs. 4a-4d; page 13, lines 8<sup>+</sup>). This request is waiting on a response from a second requester of the computing environment, and the response is to be serviced by a thread pool selected from a set of one or more eligible thread pools (page 13, lines 24<sup>+</sup>). The technique further includes, upon receipt of the request waiting on the response, and without input from the first requester or the second requester of which thread pools can service the response, dynamically altering the set of one or more eligible thread pools to provide an altered thread pool set of eligible thread pools, wherein a thread pool of the altered thread pool set is to service the response to avoid a deadlock with the request awaiting the response (page 14, lines 7-23; page 19, line 24 – page 20, line 6; page 22, lines 5-15; page 24, lines 13-25). The dynamically altering comprises setting a pool mask to indicate the eligible thread pools of the altered thread pool set to service the response (page 14, lines 7-23).

### **Issues**

1. Whether claims 1, 27, 53 and 58 contain subject matter not described in the specification, and therefore, were properly rejected under 35 U.S.C. §112, first paragraph.

2. Whether claims 1, 27, 53 and 58 are indefinite for failing to particularly point out and distinctly claim the subject matter which Appellants regard as the invention, and therefore, were properly rejected under 35 U.S.C. §112, second paragraph.

3. Whether claims 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79 were rendered obvious under 35 U.S.C. §103(a) to one of ordinary skill in the art by Schoening in view of Belkin.

### **Grouping of Claims**

Since each ground of rejection provides a grouping of claims, the following group of claims is included herein:

- I. Claims 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79

Appellants respectfully submit that the claims of Group I do not stand or fall together. For example, claims 4, 9-11, 30, 35-37, 61 & 66-68 each include additional features that provide separate basis of patentability

### **Argument**

#### **I. Claims 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79**

#### **35 U.S.C. §112:**

As noted, claims 1, 27, 53 & 58 stand rejected under 35 U.S.C. §112, first paragraph, as containing subject matter not described in the specification. Reversal of this rejection is respectfully requested.

A decision of whether an invention has been sufficiently enabled requires determination of “whether one reasonably skilled in the art could make or use the invention from disclosures in the patent coupled with information known in the art without undue experimentation.” *United States v. Telectronics, Inc.*, 827 F.2d 778, 785, 8 USPQ2d 1217, 1223 (Fed. Cir. 1988). Further, a patent need not teach, and preferably omits, what is well known in the art. *In re Buchner*, 929 F.2d 660, 661, 18 USPQ2d 1331, 1332 (Fed. Cir. 1991); *Hybritech, Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 1384, 231 USPQ 81, 94 (Fed. Cir. 1986), *cert. denied*, 480 U.S. 947 (1987); and *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1463, 221 USPQ2d 481, 489 (Fed. Cir. 1984).

If a statement of utility in the specification contains within it a connotation of how to use, and/or the art recognizes that standard modes of administration are known and contemplated, 35 U.S.C. §112 is satisfied (emphasize added). *In re Johnson*, 282 F.2d 370, 373, 127 USPQ 216,

219 (CCPA 1960); *In re Hitchings*, 342 F.2d 80, 87, 144 USPQ 637, 643 (CCPA 1965); and *In re Brana*, 51 F.2d 1560, 1566, 34 USPQ2d 1437, 1441 (Fed. Cir. 1993).

Moreover, the Manual of Patent Examining Procedure (MPEP) §2164.04 states that a “specification disclosure which contains a teaching of the manner and process of making and using an invention in terms which correspond in scope to those used in describing and defining the subject matter sought to be patented must be taken as being in compliance with the enablement requirement of 35 U.S.C. §112, first paragraph, unless there is a reason to doubt the objective truth of the statements contained therein which must be relied on for enabling support”.

Appellants respectfully submit that both judicial decisions and the MPEP are counter to the Examiner’s position with respect to the adequacy of the disclosure of the present invention. Further, the Examiner has not shown a reasonable basis for questioning the adequacy of the disclosure to enable a person of ordinary skill in the art to make and use the claimed invention. The specification is in compliance with the enablement requirement of 35 U.S.C. §112, first paragraph, since the specification discloses a process of making and using the invention in terms which correspond in scope to those used in describing and defining the subject matter sought to be patented. For these reasons, Appellants respectfully request reversal of the rejection to the claims presented herewith.

In the Office Action, the Examiner alleges that “without input from the first requestor or the second requestor” is not enabled by “without human intervention and/or client code”, page 14, lines 16-17 in the specification since the requestors are not mentioned, and a requestor/client process or client sub-process is not a user as taught by “human intervention”. This characterization of sufficiency of Appellants’ specification is respectfully traversed.

As noted in Appellants' prior response dated February 19, 2004, at page 11, Appellants indicate that support for the claim amendments can be found throughout the application. Various specific pages are then cited by way of example. The Office Action relies upon only three lines of these citations in stating the rejection, which is believed to be in error.

Appellants submit that one of ordinary skill in the art could make and use the invention from the disclosure in the specification, and that the pending claims are fully supported by the application as filed. In Appellants' independent claims, a "first requestor of the computing environment" provides a request to be processed. This request waits on a response from a "second requestor of the computing environment". By way of further example, claim 14 recites that the receiving of the request is at a server of the computing environment, and that the first requestor is a first client and the second requestor is a second client. The specification is replete with specific examples of the first requestor making a request which requires a response from a second requestor. Appellants' independent claims then recite specific functionality for dynamically altering one or more eligible thread pools to provide an altered thread pool set of eligible thread pools, wherein the altered thread pool set is to service the response to avoid deadlock with the request awaiting the response.

At page 15, lines 10-23, Appellants describe a client request (also referred to as an SMB request or data) being received by a server, step 406, and that responsive thereto SRB code is run to dispatch the request to an available thread from an eligible pool. The SRB code does not do much processing, for instance, it does not examine the SMB (i.e., client request) to see what it is. Since the receiving logic does not examine the request to determine the nature of the request, there is no input from the first requestor (or, similarly, the second requestor) regarding which thread pools can service the response. Thus, Appellants' receiving logic functions to schedule thread pools without evaluating the nature of the request itself, i.e., without input from the first requestor or the second requestor.

By way of further example, Appellants state at page 24, lines 12-25, that the invention "handles thread pool assignment dynamically with no indication from the client request which pool it can use." The "client request" in this sentence refers to requests/responses from either the



first requestor or the second requestor, as is clear to one of ordinary skill in the art from the preferred embodiments of the invention described in the specification.

To summarize, Appellants submit that the adequacy of the disclosure of the present invention is supported by both judicial decisions and the MPEP, as well as by the level of understanding of a person of ordinary skill in the art. Based on the foregoing, reversal of the 35 U.S.C. §112, first paragraph rejection to claims 1, 27, 53 & 58 is respectfully requested.

Claims 1, 27, 53 and 59 were also rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellants regard as the invention. The Examiner alleges that the “requestors that do not provide input” is unclear in combination with the specification’s teaching of “without” “client code”. Also, the Examiner alleges that it is unclear how a deadlock is avoided, and that a deadlock is present in the claims.

Regarding the requirement for claims to particularly point out and distinctly claim the invention, MPEP §2171 requires a claim to be “evaluated in the context of whether the claims is definite – i.e., whether the scope of the claims is clear to a hypothetical person possessing the ordinary level of skill in the pertinent art.” MPEP §2173.02 states that claims directed to patentable subject matter should be allowed if they define the patentable subject matter “with a reasonable degree of particularity and distinctness” (emphasis in original). MPEP §2173.02 further states:

Definiteness of claims language must be analyzed, not in a vacuum, but in light of:

- (A) The content of the particular application disclosure;
- (B) The teachings of the prior art; and
- (C) The claim interpretation that would be given by one possessing the ordinary level of skill in the pertinent art at the time the invention was made.

A person of ordinary skill in the art would interpret Appellants' claim language of "without input from said first requestor or said second requestor of which thread pools can service the response" in light of the specification examples provided. As recited in the independent claims, the "first requestor" comprises a requestor from which a request to be processed is received. Claim 14 makes it clear that the first requestor in one example is a first client of the computing environment. Because the claims themselves further characterize the requestor, in one example, as being clients, and since the specification discusses various embodiments of the invention relative to requests from clients, Appellants respectfully submit that the "first requestor" would be understood by one skilled in the art from the specification provided. As noted above, support for the functionality at issue recited in the independent claims can be found throughout the application as filed. Various lines of pages 15 & 24 are discussed briefly above, however, the examples provided make it clear that the first requestor from which the request is received does not provide an indication of which pool can be used to service the request. This functionality is recited to define the environment of the invention and distinguish the environment from other approaches wherein requests are typically accompanied with information to use a specific pool for processing. The "second requestor" is defined in the independent claims to be the requestor of the computing environment from which a response to the request is required in order to process the request. This "response" is a type of request which must itself be processed, or serviced, and that this servicing again is without input from the second requestor of which thread pool to service the response. Thus, in Appellants' invention, the dynamically altering the pools occurs within the computing environment upon receipt of the request waiting on the response, and without input from the first requestor or the second requestor of which thread pools can service the response. Since the first requestor and second requestor are defined in the independent claims, Appellants respectfully submit that the dynamically altering is recited to proceed without their input, which does not mean that there is not input from another entity of the computing environment, such as (in one example) a server of the computing environment receiving the request and awaiting the response.

Regarding "how a deadlock is avoided" and "that a deadlock is present in the claims", Appellants respectfully submit that a person of ordinary skill in the art would interpret the claims at issue in light of the specification. In the specification, Appellants subscribe that the first client

sends a request to a server to be processed, and that certain requests require a callback to other clients, which must then wait for the response to the callback to be processed before the first request can be fully processed. Since there are a finite number of service threads, Appellants provide a mechanism to ensure that responses get dispatched and run, otherwise, a deadlock could exist since if a response is never serviced, a request is never fully processed and that client that sends the request waits indefinitely for the response to the request, thus resulting in deadlock. Since the scenario is clear from the specification, and since Appellants' independent claims each discuss a request from a first requestor waiting on a response from a second requestor, both of which are to be serviced by a set of one or more eligible thread pools, Appellants respectfully submit that the potential deadlock condition is clear, and that Appellants' technique for avoiding the deadlock, by dynamically altering the one or more eligible thread pools to provide the altered thread pool set, wherein one thread pool of the altered thread pool set is to service the response to avoid the deadlock with a requestor awaiting the response would be understood by one skilled in the art as being the mechanism by which deadlocks are avoided. The dynamically altering is recited to comprise setting a pool mask to indicate the eligible thread pools of the altered thread pool set to service the response, upon which the request is waiting.

Since the independent claims are believed to define the invention with a sufficient degree of particularity and clarity for one of ordinary skill in the art, reversal of the 35 U.S.C. §112, second paragraph rejection is respectfully requested.

35 U.S.C. §103(a)

As noted, claims 1, 3-4, 9-11, 14-18, 21-22, 27, 29-30, 35-37, 40-44, 47-48, 53-56, 58, 60-61, 66-68, 70-75, and 78-79 stand rejected under 35 U.S.C. §103(a) as obvious over Schoening in view of Belkin. Reversal of this rejection is respectfully requested.

Appellants' invention is directed to the management of thread pools to service a request from a requester (e.g., a client) and a response (e.g., callback response) from another requester on which the request is waiting. For instance, a client request for a locked data file waits for another client's callback response that unlocks the data file. This thread pool management technique avoids deadlock between the request and the response on which the request is waiting

by dynamically altering an eligible thread pool set (e.g., containing a primary thread pool) to provide an altered set of thread pools. The altered thread pool set includes, for example, the primary thread pool and a secondary thread pool to service the response. By allowing, for example, the secondary thread pool to service the response and providing the primary thread pool to service the request waiting on the response, deadlock is avoided. This dynamic altering scheme is efficient because it does not require any input from the requesters that issue the request or the response. The thread pool management technique claimed uses low-level operating system functionality that is incapable of having knowledge of client input.

Appellants respectfully submit that at least the features of dynamically altering the set of one or more eligible thread pools without input from the first requester or second requester, and wherein a thread pool of the altered thread pool set is to service the response to avoid deadlock, are not taught or suggested by Schoening and Belkin, alone or in combination.

Schoening describes determining whether or not work can be run in parallel in a multi-threaded environment. When parallel processing is used, Schoening dispatches the work onto separate threads based on a partial order determined by preconditions and resource requirements associated with execution components. (see Abstract and Col. 4, lines 1-29 thereof). This parallel processing technique is different from the protocol of the present invention.

For example, Appellants' claim, in part, dynamically altering the set of one or more eligible thread pools without input from the first requester or the second requester of which thread pools can service the response. In contrast, Schoening determines, with input, which thread pools can service a response. The input used by the Schoening patent in its determination of thread pools is the partial order (e.g., evaluation sequence), which is "declared or stored in a Partial Order object and is passed as a parameter to the EvalGroup" (col. 41, lines 4-5). As depicted in FIG. 5A of Schoening, parameters associated with the evaluation sequence or partial order are passed to EvalGroup 502 from Service Module Functions (SMFs) 512 (see also col. 4, lines 6-7). SMFs receive such parameters from clients subsequent to clients acquiring access to service modules (see steps 418, 422 & 424 of FIG. 4A thereof). These clients providing the parameters are requestors as defined in Appellants' independent claims.

Further, Appellants' claim dynamically altering the set of one or more eligible thread pools (without input from the first or second requesters) to provide an altered thread pool set of eligible thread pools, wherein a thread pool of the altered thread pool set is to service the response to avoid a deadlock with the request awaiting the response. As used in the present application, "deadlock avoidance" means avoiding an indefinite wait, since if the response is never processed, there is a type of deadlock with the request waiting on the response. In contrast, Schoening does not discuss a deadlock avoidance scheme at all. Instead, Schoening describes the existence of deadlock in the case of synchronizing parallel threads (col. 3, lines 24-30) and the detection of deadlock associated with the transaction processing functions of the Asynchronous Network Interface (ANI). Although the existence and detection of deadlock is disclosed by Schoening, Appellants respectfully submit that Schoening does not suggest or imply the avoidance of deadlock with a request awaiting a response using a protocol as claimed in the present invention.

In support of the rejection of the independent claims, the Office Action stated that Schoening teaches altering thread pools and cited col. 40, lines 61-62; col. 41, lines 3-14 & 39-42; and col. 42, lines 22-24. These sections of Schoening describe using transactions to synchronize threads; passing a partial order as a parameter to EvalGroup and evaluating that partial order; processing an execution manager subsystem object (EvalGroup) in parallel with other objects; and providing the partial order of SMFs. To the extent these cited sections are deemed applicable to the claims presented herewith, Appellants traverse any conclusion that they teach or suggest the above-noted features of Appellants' claimed invention. As noted, the passing of the partial order parameter (e.g., the evaluation sequence) indicates a thread pool determination using input from requesters of which thread pool can service a request, which differs from Appellants' recited dynamic altering of the set of one or more eligible thread pools without input from the first or second requesters of which thread pools can service the response.

To summarize, Schoening fails to teach or suggest at least Appellants' recited features of (1) dynamically altering the set of one or more eligible thread pools without input from the first requester or second requester of which thread pools can service the response; and (2) the dynamically altering providing an altered thread pool set, wherein a thread pool of the altered

thread pool set is to service the response to avoid a deadlock with the request awaiting the response. Appellants respectfully submit that Belkin does not overcome these deficiencies of Schoening as applied against their claims.

For example, Belkin determines which thread pool a request is to be associated with input from a request (i.e., from a requester). In particular, Belkin describes in the Abstract thereof:

. . . When a request is received, it is processed to determine with which thread pool the request is to be associated. This processing is carried out by determining the type of service being requested by the request, and then determining which thread pool is associated with that type of service. Alternatively, this processing is carried out by extracting a set of indication information (e.g., a universal resource identifier) from the request, and then determining which thread pool is associated with that set of indication information . . .

Belkin also discloses that the determination of the thread pool utilizes user-defined tables. For example, Belkin states that “with tables 200 and 300a, the user can freely define any thread pool, and associate any type of service with any defined thread pool” (col. 7, lines 44-46; see also FIGs. 2 & 3 thereof). Moreover, relative to user-defined table 200, Belkin describes an evaluation function to determine which thread pools are appropriate to service a request (col. 15, lines 29-44). This evaluation function’s determination is “user-provided” and “based upon the request” (col. 15, lines 32-36). Thus, the determination in Belkin of which thread pool to use is performed with input from a user (i.e., client or requestor (see FIG. 1 thereof)) of which thread pool is to service a request, which is clearly different from a process which dynamically alters a set of one or more eligible thread pools without input from a first requester or a second requester of which thread pools can service the response, as recited in the claims presented herewith.

Further, Belkin describes detection of deadlock when a thread pool has no free threads available (col. 16, lines 3-5). Belkin’s detection of deadlock is directed to invoking the evaluation function to order requests on a queue associated with the thread pool (col. 16, lines 14-43). The invocation of the evaluation function in Belkin does not address deadlock avoidance, per se, as claimed by the present invention. Instead, it provides the dispatching code in Belkin with input regarding the thread pool on which the request is to be placed, and where to

put the request on the queue. Moreover, this deadlock-prompted action associated with input of which thread pool can service a request is different from the deadlock avoidance recited by the claims presented herewith, wherein a thread pool of the altered thread pool set is to service the response to avoid a deadlock with the request awaiting the response, and the altered thread pool set is provided by the dynamic alteration of the set of one or more eligible thread pools without input from the first requester or the second requester of which thread pools can service the response.

In the final Office Action, it is stated that Belkin teaches dynamically determining different eligible thread groups, and col. 4, lines 38-39; col. 7, lines 41-42; col. 15, lines 33-35 and col. 16, lines 44-46 of Belkin are cited. These sections disclose implementing multiple thread pools; specifying, by means of a row of a table, an association between a particular thread pool and a particular type of service; invoking the user-provided evaluation function to evaluate a request; and invoking the evaluation function by a request processing mechanism. Appellants respectfully traverse any conclusion that these cited sections of Belkin teach or suggest the subject matter of their claims. As noted, the invocation of the evaluation function based on a request and the user-defined associations in the table provide input of which thread pools can service a request, which is in contrast to the dynamic alteration without input from the first or second requesters of which thread pools can service a response, as recited in Appellants' claims.

Since both Schoening and Belkin fail to teach or suggest Appellants' claimed protocol of (1) dynamically altering the set of one or more eligible thread pools without input from the first requester or second requester of which thread pools can service the response; and (2) the dynamically altering providing an altered thread pool set, wherein a thread pool of the altered thread pool set is to service the response to avoid a deadlock with the request awaiting the response, it is respectfully submitted that the combination does not render their invention obvious.

For the above reasons, Appellants respectfully request reversal of the obviousness rejection of independent claims 1, 27, 53 & 58. The dependent claims are believed patentable for the same reasons as the independent claims from which they directly or ultimately depend, as

well as for their own additional characterizations. For example, claims 4 & 9-11 (as well as the corresponding system claims 30 & 35-37 and program storage device claims 61 & 66-68) are believed to recite separate basis for patentability.

Claim 4 repeats the subject matter of claim 1, and further adds the characterization that the dynamically altering is initiated when it is determined that the request is waiting for the response. For an alleged teaching of this protocol, the Office Action references Col. 39, lines 65-67 of Schoening. These lines state: "It is known that when executing Service Module Function that involves communicating to the devices, the function must often wait while the external devices complete some other function." Appellants respectfully submit that there is no suggestion in this discussion of Schoening for the process recited by Appellants in claim 4, wherein the dynamically altering (discussed above) is initiated when it is determined that the request is waiting for the response. Initiation of a dynamic altering of a thread pool set based upon this condition determination is not taught or suggested by Schoening.

Claim 9 again repeats the subject matter of claim 1, and further specifies protocol which includes dynamically re-altering the altered thread pool set to service one or more other responses or one or more other requests. Claim 10 further recites the subject matter of claim 9 and indicates that the dynamic re-altering is performed when there are no outstanding callbacks to be responded to by the second requester, while claim 11 specifies the subject matter of claim 10 and further qualifies the process by reciting determining whether there are any outstanding callbacks, wherein the determining references a data structure associated with the second requester. No similar processing is believed taught or suggested by Schoening, including at Col. 40, lines 36-38, Col. 15, lines 46-48 and Col. 16, lines 7-8 thereof cited in the final Office Action. The Federal Circuit has expressly mandated that functional claim language be considered in evaluating a claim relative to the prior art. Appellants respectfully submit that the application of this standard to the claims at issue leads to the conclusion that the recited subject matter would not have been obvious to one of ordinary skill in the art based upon the applied patents.

For the above reasons, Appellants respectfully request reversal of the obviousness rejection to all claims of Group I.



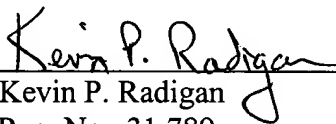
### Conclusion

Appellants herein request reversal of all rejections set forth in the final Office Action. Appellants respectfully submit that the adequacy of the disclosure of the present invention is supported by both judicial decisions and the MPEP, as well as by the level of understanding of a person of ordinary skill in the art. Additionally, Appellants submit that the claims presented particularly point out and distinctly claim the subject matter which Appellants regard as the present invention. Further, Appellants respectfully submit that their claimed invention would not have been rendered obvious by Schoening in view of Belkin. The art does not, individually or in combination, teach or imply Appellants' invention which includes, in part, dynamically altering the set of one or more eligible thread pools without input from the first requester or second requester of which thread pools can service the response; and the dynamically altering providing an altered thread pool set, wherein a thread pool of the altered thread pool set is to service the response to avoid a deadlock with the request awaiting the response.

For at least the above reasons, Appellants allege error in rejecting the recited invention.

Accordingly, reversal of all rejections is respectfully requested.

Respectfully submitted,



Kevin P. Radigan  
Reg. No. 31,789  
Attorney for Appellants

Dated: August 31, 2004

HESLIN ROTHENBERG FARLEY & MESITI, P.C.  
5 Columbia Circle  
Albany, New York 12203  
Telephone: (518) 452-5600  
Facsimile: (518) 452-5579

## **Appendix**

1. A method of managing thread pools of a computing environment, said method comprising:

receiving from a first requester of said computing environment a request to be processed, said request waiting on a response from a second requester of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

upon receipt of the request waiting on the response, and without input from said first requester or said second requester of which thread pools can service the response, dynamically altering said set of one or more eligible thread pools to provide an altered thread pool set of eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

wherein said dynamically altering comprises setting a pool mask to indicate said eligible thread pools of said altered thread pool set to service said response.

3. The method of claim 1, wherein said pool mask is included within a data structure associated with said response.

4. The method of claim 1, wherein said dynamically altering is initiated when it is determined that said request is waiting for said response.

9. The method of claim 1, further comprising dynamically re-altering said altered thread pool set to service one or more other responses or one or more other requests.

10. The method of claim 9, wherein said dynamically re-altering is performed when there are no outstanding callbacks to be responded to by said second requester.

11. The method of claim 10, further comprising determining whether there are any outstanding callbacks, said determining referencing a data structure associated with said second requester.

14. The method of claim 1, wherein said receiving comprises receiving said request by a server of said computing environment, and wherein said first requester is a first client and said second requester is a second client.

15. The method of claim 14, wherein said server is a file server.

16. The method of claim 14, wherein at least one of said first client and said second client runs on a same physical computer of said computing environment as said server.

17. The method of claim 14, wherein at least one of said first client and said second client runs on a different physical computer of said computing environment than said server.

18. The method of claim 1, wherein said dynamically altering is performed by a server of said computing environment.

21. The method of claim 1, wherein said first requester and said second requester are the same requester.

22. The method of claim 1, wherein said first requester and said second requester are different requesters.

27. A system of managing thread pools of a computing environment, said system comprising:

means for receiving from a first requester of said computing environment a request to be processed, said request waiting on a response from a second requester of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

means for, upon receipt of the request waiting on the response, and without input from said first requester or said second requester of which thread pools can service the response, dynamically altering said set of one or more eligible thread pools to provide an altered thread pool set of eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

wherein said means for dynamically altering comprises means for setting a pool mask to indicate said eligible thread pools of said altered thread pool set to service said response.

29. The system of claim 27, wherein said pool mask is included within a data structure associated with said response.

30. The system of claim 27, wherein the dynamically altering is initiated when it is determined that said request is waiting for said response.

35. The system of claim 27, further comprising means for dynamically re-altering said altered thread pool set to service one or more other responses or one or more other requests.

36. The system of claim 35, wherein said means for dynamically re-altering is performed when there are no outstanding callbacks to be responded to by said second requester.

37. The system of claim 36, further comprising means for determining whether there are any outstanding callbacks, said determining referencing a data structure associated with said second requester.

40. The system of claim 27, wherein said means for receiving comprises means for receiving said request by a server of said computing environment, and wherein said first requester is a first client and said second requester is a second client.

41. The system of claim 40, wherein said server is a file server.

42. The system of claim 40, wherein at least one of said first client and said second client runs on a same physical computer of said computing environment as said server.

43. The system of claim 40, wherein at least one of said first client and said second client runs on a different physical computer of said computing environment than said server.

44. The system of claim 27, wherein the dynamically altering is performed by a server of said computing environment.

47. The system of claim 27, wherein said first requester and said second requester are the same requester.

48. The system of claim 27, wherein said first requester and said second requester are different requesters.

53. A system of managing thread pools of a computing environment, said system comprising:

a processor adapted to receive from a first client of said computing environment a request to be processed, said request waiting on a response from a second client of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

said processor being adapted to, upon receipt of the request waiting on the response, and without input from said first client or said second client of which thread pools can service the response, dynamically alter said set of one or more eligible thread pools to provide an altered thread pool set of eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

said processor being adapted to set a pool mask to indicate said eligible thread pools of said altered thread pool set to service said response.

54. The system of claim 53, wherein said processor comprises a server of said computing environment.

55. The system of claim 53, wherein said first client and said second client are the same client.

56. The system of claim 53, wherein said first client and said second client are different clients.

58. At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of managing thread pools of a computing environment, said method comprising:

receiving from a first requester of said computing environment a request to be processed, said request waiting on a response from a second requester of said computing environment, and wherein said response is to be serviced by a thread pool selected from a set of one or more eligible thread pools;

upon receipt of the request waiting on the response, and without input from said first requester or said second requester of which thread pools can service the response, dynamically altering said set of one or more eligible thread pools to provide an altered thread pool set of eligible thread pools, wherein a thread pool of the altered thread pool set is to service said response to avoid a deadlock with said request awaiting said response; and

wherein said dynamically altering comprises setting a pool mask to indicate said eligible thread pools of said altered thread pool set to service said response.

60. The at least one program storage device of claim 58, wherein said pool mask is included within a data structure associated with said response.

61. The at least one program storage device of claim 58, wherein said dynamically altering is initiated when it is determined that said request is waiting for said response.

66. The at least one program storage device of claim 58, wherein said method further comprises dynamically re-altering said altered thread pool set to service one or more other responses or one or more other requests.

67. The at least one program storage device of claim 66, wherein said dynamically re-altering is performed when there are no outstanding callbacks to be responded to by said second requester.

68. The at least one program storage device of claim 67, wherein said method further comprises determining whether there are any outstanding callbacks, said determining referencing a data structure associated with said second requester.

70. The at least one program storage device of claim 69, wherein said ordering comprises having a primary thread pool selectable before any secondary thread pool.

71. The at least one program storage device of claim 58, wherein said receiving comprises receiving said request by a server of said computing environment, and wherein said first requester is a first client and said second requester is a second client.

72. The at least one program storage device of claim 71, wherein said server is a file server.

73. The at least one program storage device of claim 71, wherein at least one of said first client and said second client runs on a same physical computer of said computing environment as said server.

74. The at least one program storage device of claim 71, wherein at least one of said first client and said second client runs on a different physical computer of said computing environment than said server.

75. The at least one program storage device of claim 58, wherein said dynamically altering is performed by a server of said computing environment.

78. The at least one program storage device of claim 58, wherein said first requester and said second requester are the same requester.

79. The at least one program storage device of claim 58, wherein said first requester and said second requester are different requesters.

\* \* \* \* \*



**TRANSMITTAL OF APPEAL BRIEF (Large Entity)**Docket No.  
EN999058

In Re Application Of: Doolittle et al.

SEP 02 2004

Application No.  
09/443,202Filing Date  
11/18/1999Patent Examiner  
Stephan F. WillettCustomer No.  
30400Group Art Unit  
2141Confirmation No.  
6901

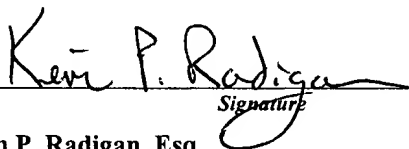
Invention: **METHOD, SYSTEM AND PROGRAM PRODUCTS FOR MANAGING THREAD POOLS OF A COMPUTING ENVIRONMENT TO AVOID DEADLOCK SITUATIONS BY DYNAMICALLY ALTERING ELIGIBLE THREAD POOLS**

COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on July 2, 2004

The fee for filing this Appeal Brief is: \$330.00

- ☐ A check in the amount of the fee is enclosed.
- ☐ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 09-0457 (IBM)

  
Signature

Kevin P. Radigan, Esq.  
Registration No.: 31,789

**HESLIN ROTHENBERG FARLEY & MESITI, P.C.**  
5 Columbia Circle  
Albany, New York 12203  
Tel: (518) 452-5600  
Fax: (518) 452-5579

Dated: August 31, 2004

I certify that this document and fee is being deposited on August 31, 2004 with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

  
Signature of Person Mailing Correspondence**Kevin P. Radigan**

Typed or Printed Name of Person Mailing Correspondence

CC: